Displaying Web Content in iPhone Applications

UIWebView

Session 124

These are confidential sessions—please refrain from streaming, blogging, or taking pictures.

Displaying Web Content in iPhone Applications

UIWebView

Wil Turner
Developer Technical Support

UIWebView

Displaying rich text and web content in your application

In This Session:

- What is UIWebView?
- Using UIWebView
- Best practices
- Advanced topics
- What's new in iPhone OS 3.0

What Is UIWebView?

Rich content in your application

The Challenge

Text presentation on the iPhone

- The basics:
  - UILabel
  - UITextField
  - UITextView
  - NSString (UIStringDrawing)
- Single font
- Single color

- Fine for many uses, but sometimes you just need style…

The Challenge

My document—I just want to look at it!

- Need to display documents
  - PDF
  - iWork
  - MS Office

- Being able to view these documents is a must…but how?

The Challenge

A World Wide Web of content

- The Internet has immense resources that can enhance your app

- Option: Exit to Safari

- Not the greatest user experience…

The Challenge

I already have this great web app…

- But it can’t access native functionality
  - UIKit
  - Address book
  - Accelerometer
  - MapKit
  - And more…

- You don’t want to start over, you want to enhance what you have…

The Solution: UIWebView

UIView

variety of content types

asic browsing

y WebKit

WebKit

A rendering engine

- Content types:
  - HTML
    - CSS
    - JavaScript
  - PDF

- Want styled text in your app? Just use HTML

UIWebView Adds Documents

- UIWebView converts to HTML
- WebKit renders
- UIWebView displays the results
- Supported types include
  - iWork ’06 and newer
  - MS Office ‘97 and newer
  - RTF and RTFD

UIWebView, the Solution

When to use UIWebView

- Displaying styled content
- Showing document previews
- Embedding web resources
- Extending web apps with native functionality

In This Session:

- What is UIWebView?
- Using UIWebView
- Best practices
- Advanced topics
- What's new in iPhone OS 3.0

Using UIWebView

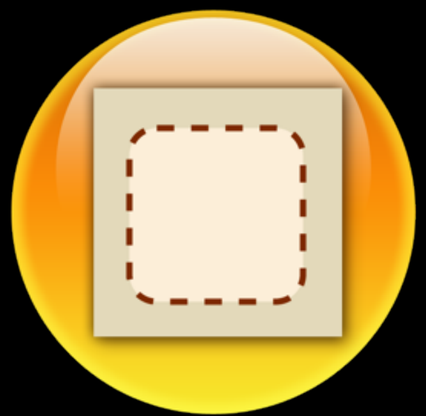
Putting UIWebView to work

Design Patterns

The players

UIWebView

UIView subclass that renders web content



<UIWebViewDelegate>

Controller object in your application
Tracks content loading in the web view

Design Patterns

Supporting roles

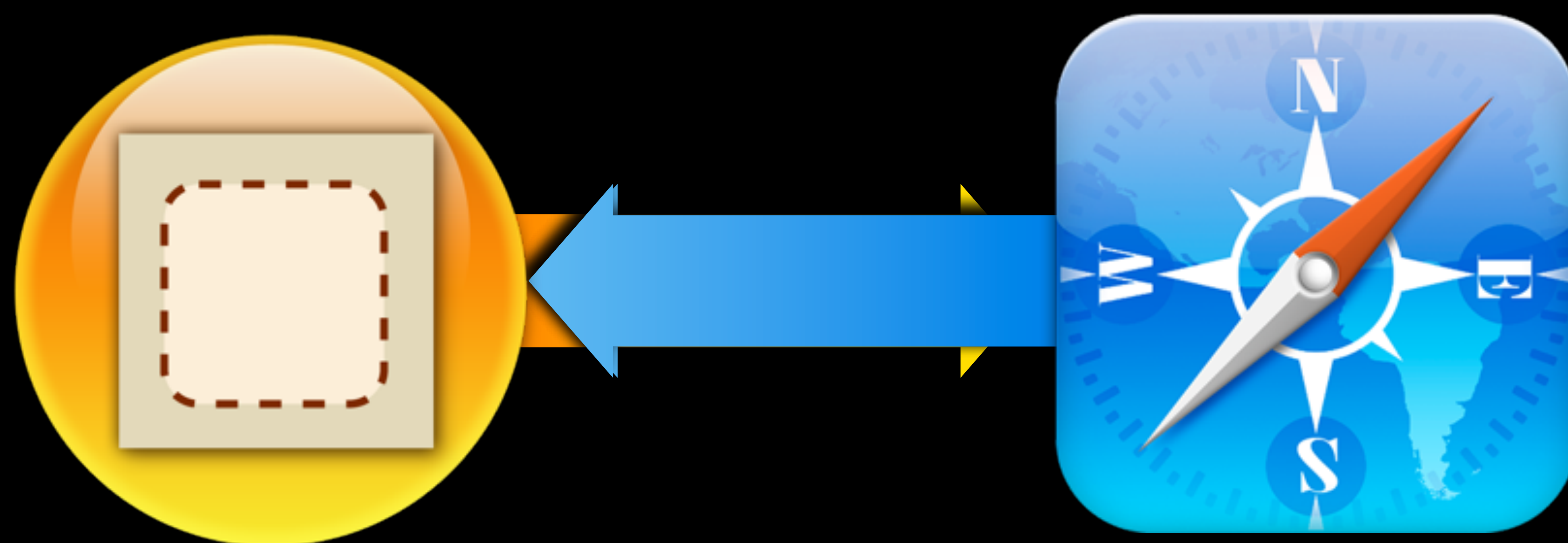
URLs

HTML

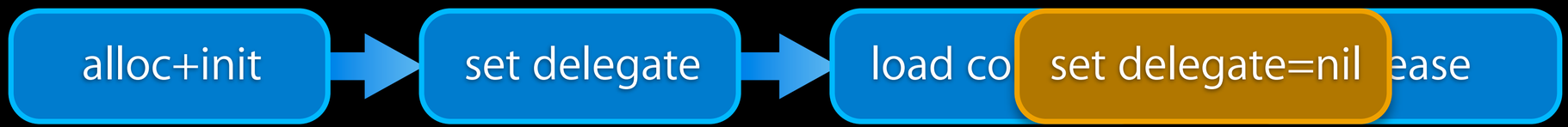
JavaScript

How They Interact

[illegible]
(handle errors here)



Life Cycle

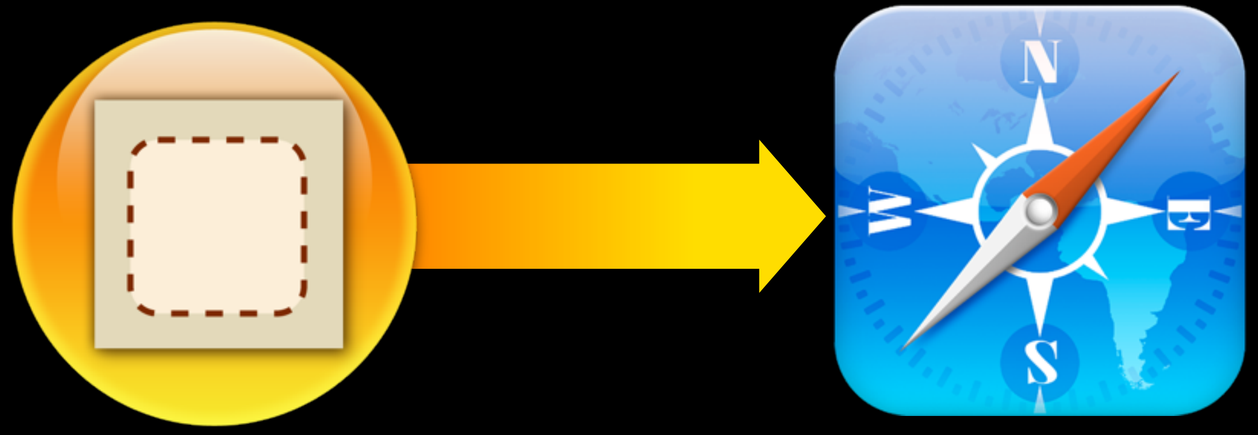


Setting the web view delegate to nil before releasing is essential!

Communicating with Content

From Objective-C to JavaScript

- How can native code trigger JavaScript?



stringByEvaluatingJavaScriptFromString:

Communicating with Content

From JavaScript to Objective-C

- When would you want to do this?
- Native functionality in AJAX callbacks
- No explicit API, but…
- Hook into the delegate method:

```
webView:shouldStartLoadWithRequest:navigationType:
```

Communicating with Content

From JavaScript to Objective-C

- Create a URL with a dummy host
- Use the URL path to specify a command
  - http://my_javascript/my_command
- In JavaScript, "load" a URL with this host:

```
function invoke_objective_c() {
    window.location = "http://my_javascript/my_command";
}
```

Communicating with Content

From JavaScript to Objective-C

- In the delegate, test the URL:

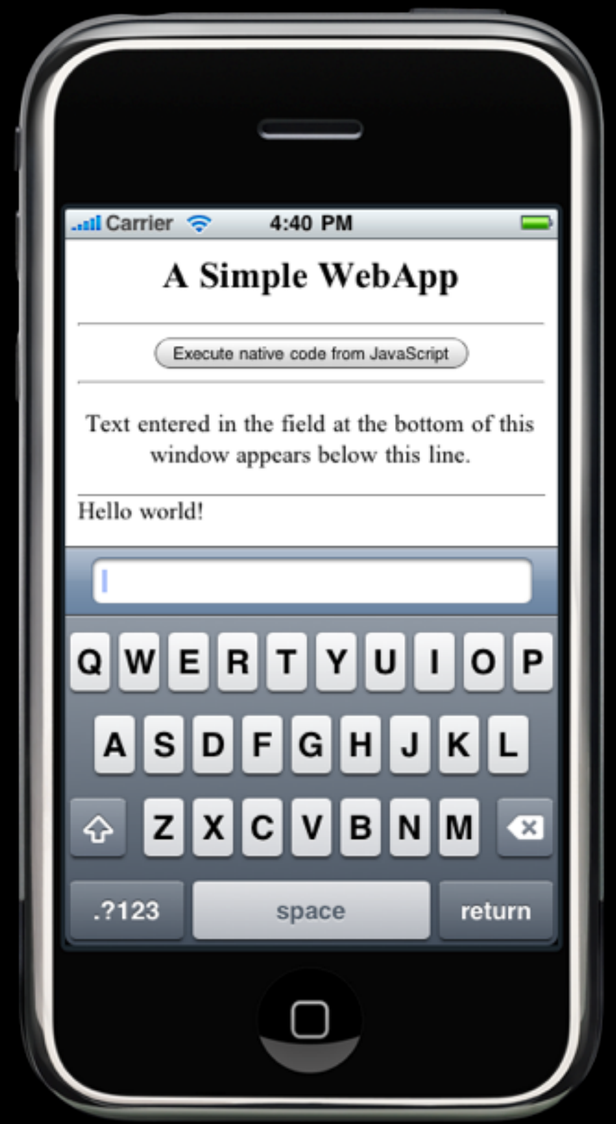
```
NSString *host = request.URL.host;
if ([host isEqualToString:@"my_javascript"]) {
    NSString *command = request.URL.path.lastPathComponent;
    if ([command isEqualToString:@"my_command"]) {
        // do something
    }
}
```

Putting UIWebView to Work

- UIWebView, UIWebViewDelegate
- Message flow
- Set delegate=nil before releasing!
- Objective-C to JavaScript
- JavaScript to Objective-C

What We'll Build:

- Embedded HTML content
- Native components
  - UIWebView
  - UIToolbar
  - UITextField
- HTML button executes native code
- Text entry executes JavaScript



Demo

Building a simple hybrid application

In This Session:

- What is UIWebView?
- Using UIWebView
- Best practices
- Advanced topics
- What's new in iPhone OS 3.0

Best Practices

Getting the most out of UIWebView

Right Tool, Right Job

Using UIWebView wisely

Rendering Engine vs. Browser

What does it take to build a complete browser?

	WebKit on Mac OS X	UIWebView on iPhone OS
Classes	125	1
Protocols	27	1
APIs	> 1000	20

UIWebView

Lightweight presentation layer over WebKit

- Optimized for displaying styled content
- Works best with known content
- Avoid general browsing

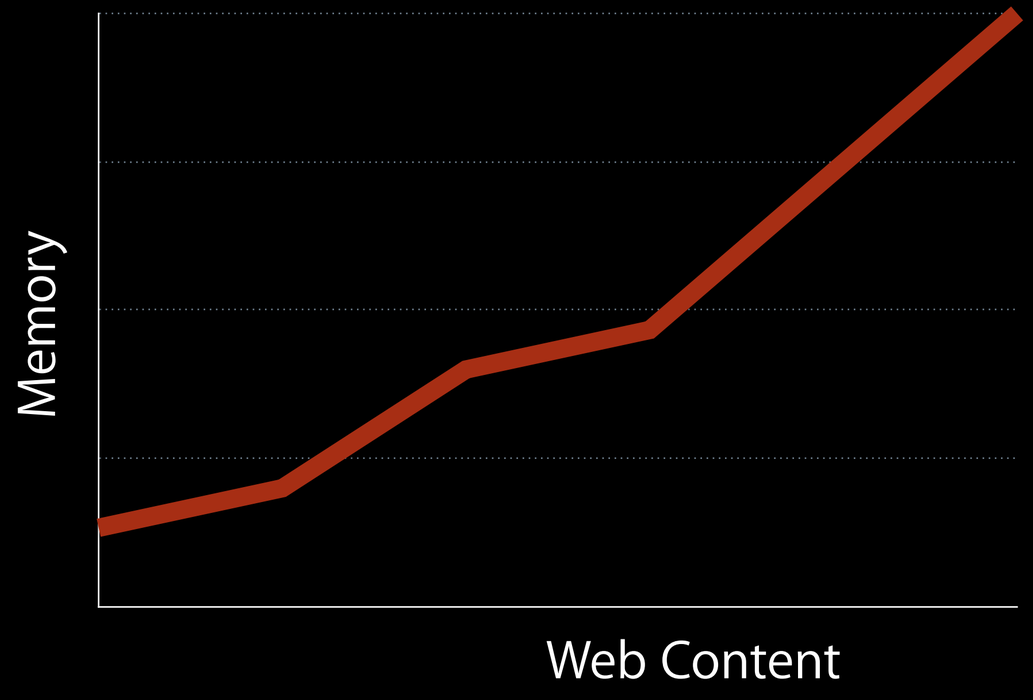
```
webView:shouldStartLoadWithRequest:navigationType:
```

Memory Consumption

- Requires significant resources
- Web pages have no size limit
- Memory consumption is dictated by content size
- Data may be cached internally

Multiple Web Views?

- More web content = more memory used



- Be conservative in your use of web views

JavaScript

- Scripts should be fast
  - Must finish within 10 seconds
  - Break longer scripts into pieces
- Power consumption
  - Network activity is expensive
  - Avoid rapid polling with AJAX
  - Where possible, batch data transfers together

Threads

- Access on main thread ONLY!
  - *Just like all other UIViews*
- Background threads are used to:
  - Download
  - Render
  - Set delegate = nil before release
- Won’t block your main thread

Getting Touch Events

Should I subclass UIWebView?

- Internal view hierarchy is PRIVATE
- Event handling is complex
  - "Forwarding" events is not supported
    - Don't directly invoke UIResponder touches methods

- Summary: Subclassing UIWebView is strongly discouraged

Getting Touch Events

A better way…

- Observe events at the window level for native logic
  - Subclass UIWindow
  - Override sendEvent:
  - Always invoke [super sendEvent:]

- Use JavaScript event handling for content-related interactions

Demo

Getting touches in a web view

In This Session:

- What is UIWebView?
- Using UIWebView
- Best practices
- Advanced topics
- What's new in iPhone OS 3.0

Advanced Topics

Pushing the envelope

Advanced Topics

- Using NSURLConnection
- Handling HTTP authentication
- Links that open in new windows
- Saving and restoring state
- Offscreen rendering

NSURLConnection

Downloading content directly

- Foundation level API
- Has synchronous and asynchronous modes
  - Synchronous connections BLOCK!
  - Use asynchronous instead
- Configure with a URL Request
- Delegate tracks loading and responses

- Fundamental for many advanced use cases…

HTTP Authentication

- You can embed authorization information in URLs
  - https://user:pass@www.myprotectedsite.com/
- Presenting a UI for authentication:
  - Use a separate URL Connection
    - connection:didReceiveAuthenticationChallenge:
    - NSURLCredentialPersistenceForSession
  - Web view will use cached credentials

Links That Open Windows

- UIWebView ignores links that open new windows
  - target=_blank
  - window.open()
- Easiest solution:
  - Don’t use this functionality in your content
- More difficult:
  - Download content
  - Modify links to open in same window
  - Then load with web view

Saving and Restoring State

- Easiest when you control content
  - Keep CSS and JavaScript inline
  - Minimize external resources
- Download content with NSURLConnection
- Load into web view
- On UIApplicationWillTerminate, save to file
- At launch, if file exists, load content

Offscreen Rendering

Creating an image representation of web content

- Load content in a web view
- UIGraphicsGetImageFromCurrentImageContext

- Challenges:
  - UIWebView won't render unless it's in a window
  - Drawing is asynchronous

Demo

Offscreen rendering

In This Session:

- What is UIWebView?
- Using UIWebView
- Best practices
- Advanced topics
- What's new in iPhone OS 3.0

What's New in iPhone OS 3.0

Welcome to the future…

What's New in iPhone OS 3.0

Copy / Paste

- Built into UIWebView
  - Standard behavior "for free"

- Disable selection by using CSS:

```
-webkit-user-select:none;
```

- Avoid universal disabling of selection
  - Only disable for specific elements

What's New in iPhone OS 3.0

Data Detectors

- Automatically convert data to clickable URLs

```
@property (nonatomic) UIDataDetectorTypes dataDetectorTypes
```

  - Phone numbers
  - Links
  - None
  - All—automatically adds future types without recompile

- Use instead of detectsPhoneNumbers property

In This Session:

- What is UIWebView?
- Using UIWebView
- Best practices
- Advanced topics
- What's new in iPhone OS 3.0

More Information

Vicki Murley
Safari Technologies Evangelist
vicki@apple.com

Documentation

UIWebView Class Reference
UIWebViewDelegate Protocol Reference
NSURLConnection Class Reference
URL Loading System
http://developer.apple.com/iphone

Related Sessions

Sophisticated Media Playback and Animation Through Open Web Standards	Mission Wednesday 3:30PM
Local Data Storage and Offline Web Applications for Safari	Russian Hill Thursday 3:30PM

Labs

iPhone Web Applications Lab	iPhone Lab A Wednesday 5:00PM
Media Playback in Web Applications Lab	OS X Lab C Thursday 10:30AM
Local Data Storage and Offline Web Applications Lab	OS X Lab A Thursday 5:00PM

Q&A

WWDC Session Survey

You can rate your sessions on the WWDC Attendee Site.

Survey forms are located on each session description page at:

developer.apple.com/wwdc/attendee

WWDC09